Économie circulaire et Écoresponsabilité des systèmes embarqués : divorce dans le couple logiciel / matériel

Fetch 2024, Maillen, Belgique

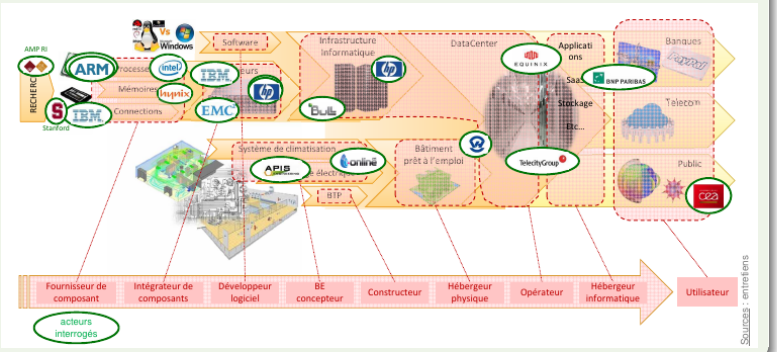
Henri-Pierre CHARLES

CEA DSCIN department / Grenoble

07 - 09 février 2024

Introduction : Data Center Value Chain

Data Center Value Chain



Missing Value : SW Stack

- Server : LAMP (Linux / Apache / PHP MySQL)
- Client : browser, Javascript engine / HTML / CSS
- Value : user data

Software has no “value” but is responsible of hardware obsolescence

Old Hardware Examples : SharpPC1500

Sharp PC-1500

- Release : 1981
- Lifespan : 4 years
- Software update : 0
- Programming language : basic

No user evolution : no obsolescence

Illustration

Introduction : Niklaus Wirth (15 February 1934, 1 January 2024)

Niklaus Wirth in 2005. Niklaus Wirth

Wirth Projects

- Pascal 1968-1972 Pascal2 / P-Code - UCSD - TurboPascal
- Modula2 1973-76
- Oberon 1977-1980
- Lilith 1977-1981

Books / Articles

- “The Pascal User Manual and Report”
- Algorithms + Data Structures = Programs
- La loi de Wirth (1995)

Introduction : Wirth's Law

Wirth's law

A Plea for Lean Software

Niklaus Wirth
ETH Zürich

Memory requirements of today's workstations typically jump substantially—from several to many megabytes—whenever there's a new software release. When demand surpasses capacity, it's time to buy add-on memory. When the system has no more extensibility, it's time to buy a new, more powerful workstation. Do increased performance and funcionality keep pace with the increased demand for resources? Mostly the answer is no.

- February 1995 DOI : 10.1109/2.348001
- "Software is getting slower more rapidly than hardware is becoming faster"
- Is the the Moore Law's to the minkind benefit ?

Illustrations

- Windows boot speed
- Android "killer" application
- Visual Studio Code
- Android Studio

Reasons : Business model

- Deliver fast, not reliable
- Tricking functionnality : not only provide a functionnality, push advertisment, get data from user,
- Add functionnality, do not choose. The text editor nightmare : where is the menu option to activate a given functionnalyty ?

Introduction : Programming Landscape

Programming Action (Web / Mobile applications)

1. Translate an action in program
2. As fast as possible : compagny cost
3. As generic as possible : improve reuse
4. Fulfill users needs

Warning implication

1. Action are sequential, not parallel
2. Code quality decrease
3. Code efficiency decrease
4. Find users to advertise

Old school programmers

1. Use your programming language
2. Use your tool
3. Sell by yourself

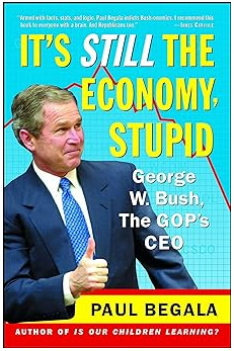
Android example

1. Mandatory programming language : Java / Kotlin
2. Android studio mandatory
3. “Store applicatif” : security versus censure (cf Apple Application ppppstore controversy)

BloatWareExamples : Android Applications

Android Application Funding

- Mainly with advertising
- Application implication
  - % of screen surface
  - % of user time



Programmer dilemna

Android Platform/API V

ANDROID PLATFORM VERSION		API LEVEL	CUMULATIVE DISTRIBUTION
4.4	KitKat	19	
5	Lollipop	21	99,6%
5.1	Lollipop	22	99,4%
6	Marshmallow	23	98,2%
7	Nougat	24	96,3%
7.1	Nougat	25	95,0%
8	Oreo	26	93,7%
8.1	Oreo	27	91,8%
9	Pie	28	86,4%
10	Q	29	75,9%
11	R	30	59,8%
12	S	31	38,2%
13	T	33	22,4%

Last updated: October 1, 2023

BloatWare Examples : Traitement Texte

Statistique des extensions des fichiers source de libreoffice 7.6.4.1.

```
  2412 :      mk (   4M)
  2605 :    java (  17M)
  5379 :     xml (  41M)
  5499 :     idl (  13M)
  9707 :     hxx (  47M)
 10656 :     cxx ( 197M)
 44928 :     svg ( 120M)
 46244 :     png (  39M)
Dir. 9072, Files 145874
```

Explications : “User friendly”

- Images : .svg .png
- Programmes : .cxx .hxx .java
- 4 fois plus d’images que de programmes

Introduction : Compiler Cheat Sheets

Classical options

cc -E Preprocessor (all # stuff : rewriting)

cc -S Compilation (from C to textual assembly)

cc -c Assembly (from textual asm to binary asm)

Executable (binary + dynamic library)

Control options

-O1 activate 50 -f option, optimize without increasing compilation time

-O2 O1 + 46 -f options, optimize, increase compilation time, no memory increase

-O3 O1 + O2 + 16 -f options, optimize, increase compilation time, memory increase

-Ofast dangerous for numerical algorithms

-Wall (60 warning options) “This enables all the warnings about constructions that some users consider questionable”

-Wextra Turn all warning as error + should be

Specific options

-march=XXXX XXX = an architecture name + optional accelerators

-march=armv8 with options +fp +simd +flagm +fp16fml +dotprod

-march=rv64imfdv zicsr zve32f zve32x zve64d zve64f zve64x zvl128b zvl32b zvl64b

–help=target,optimizers explain specific optimization

Compiler objective

- Managing complexity (IA inside !)
- Do not blindly rely on it

Introduction : Value Chain Compiler

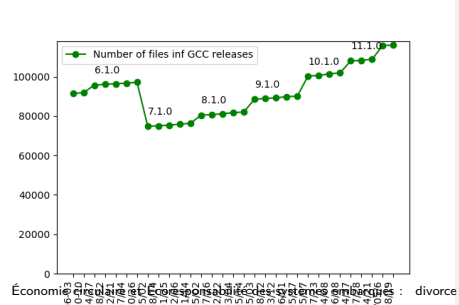
Compiler has no "value"

- GCC exist since 1987
- GNU_Compiler_Collection
- 15 million LOC (2019)
- More than 400 contributors

Industrial concern

- Intel has stopped icc compiler
- IBM has stopped xlc compiler
- Apple has invested in LLVM
- Intel & IBM invest in GCC

GCC Hardware Specific options evolution



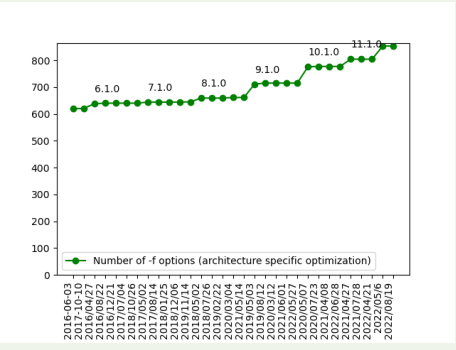
Active Research Domain

- CGO : Conf. on Code Geneation and Optimization
- HiPEAC Conf. + Journal TACO : Transaction on Achitecture and Code Optimisation)
- ESWEEK / CASES : Conf. on Compilers, Architectures, and Synthesis for Embedded Systems + Journal TCAD

Introduction : Value Chain Compiler

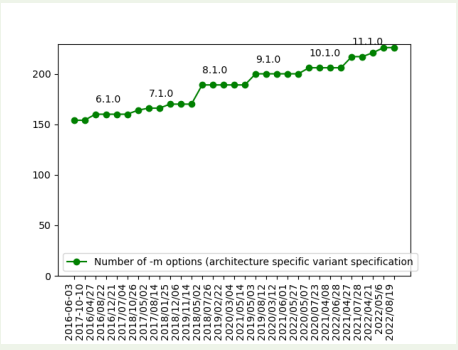
-f options number

- Activate one specific optimisation
- Application dependent



-m options number

- Activate an architecture option
- Architecture / accelerator version dependant



Introduction : Industrial Hardware Failures related relation with compilers

Itanium aka “Itanic” (Launch 2001)

- Intel & HP collaboration
- VLIW ISA + X86 ISA

The latter three were canceled before reaching the market. By 1997, it was apparent that the IA-64 architecture and the compiler were much more difficult to implement than originally thought, and the delivery timeframe of Merced began slipping.[45]

IBM CELL BE (2006) Cell (processor)

- Half sucess in Playstation 3
- 1st top500 computer “Roadrunner” to reach 1.0 petaflop
- 1st heterogeneous processor with 2 distincts ISA (POWER + SPE)

SPEs do not have any branch prediction hardware (hence there is a heavy burden on the compiler)

Solutions from Compilers : the main tool to fight obsolescence

Compiler Evolution Strategy

- Slowly remove features :
  Linux has removed Itanium support, GCC does not
- Fast integration

Follow 3 evolutions

- SW evolutions : applications
- SW evolutions : platform
- HW evolutions

At least Intel continues to be quite punctual in their open-source compiler work and getting out new ISA features and CPU family targets added to the upstream compiler code-bases well in advance of product launches. So by the time we're actually seeing AVX10.1 (or more excitingly, AVX10.2) enabled processors in the wild, all of the open-source compiler support will ideally be all sorted out and in released versions.

ref: Phoronix

Solutions : Regulation

Existing Political Regulation

- RGPD
- Apple support USB-C
- Apple to support tiers software stores
- Google deconnect

Hardware support : an unstoppable tsunami

- Popular hardware dies in techno-push evolutions
- Impossible (to costly) to rebuild MPSoC on old fab

Why not regulate

- HW lifespan for laptop / smartphones (10 years ?)
  - Fairphone : fairtrade + long lifespan
  - Long term support for basic SW (OS)
- SW same lifespan (support age)
  - LineageOS support FairPhone and old popular smartphones
- What about SW applications ?
- RISC-V ?

Solutions : HW / SW support

SW tool for HW development (already used)

- Application driven
- Multi level SW emulation
  - QEMU
  - GEM5
  - VHDL
  - ../..
- Code generation
  - Need for new compiler paradigms

SW tool for long term HW support (to be invented)

- Should integrate new HW accelerator in programming language concepts
- Run new application on old hardware. DBT : replace inexistant new instructions in the fly
- Run new application by recompilation : should target old HW / OS / HW
- ../.. something to break the tsunami ...

Conclusion / Ideas

Possible technical solutions

- Joint innovation on SW / HW
- Compilers are a already a solution for circular (invisible) economy
- Compiler / low level tools should be revisited
  - for ISA heterogeneity,
  - for explainable / understandable results

Possible industrial solutions

Business model are guilty, not the technology

- Political regulation
- Regulate SW evolution
- Regulate SW quality (lifespan)

No more “plea” a constraint

Some Technical Challenges

- RISC-V explosion : danger ahead !
- Software Innovation should be taken seriously : need new tools and strategies
- SW tools has no value in value chain but are HW enabler
- OpenSource SW should be promoted : since they have no “value” they should have a better code quality
- Application providers should be regulated
- Interoperability instead of silos aka Unix philosophy
  - Write programs that do one thing and do it well.
  - Write programs to work together.
  - Write programs to handle text streams, because that is a universal interface.